Ames Research Center

ENHANCEMENTS TO THE GRID GENERATION SCRIPT LIBRARY AND POST-PROCESSING UTILITIES IN CHIMERA GRID TOOLS

William M. Chan

NASA Ames Research Center

9th Symposium on Overset Composite Grid and Solution Technology,
State College, Pennsylvania, October 14-16, 2008

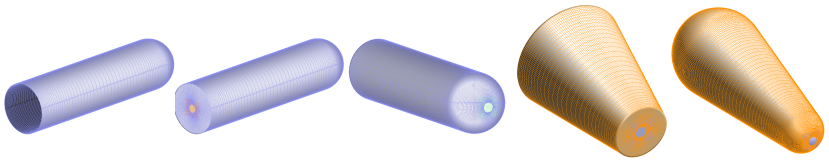
OVERVIEW

- Recent additions to the Chimera Grid Tools (CGT) script library for geometry and grid creation

- Enhancements to the hybrid mesh approach to surface loads integration on overset grids

- Introduction to OVERSMART (Solution Monitoring and Reporting Tool)

CGT SCRIPT LIBRARY: CYLINDER MACRO

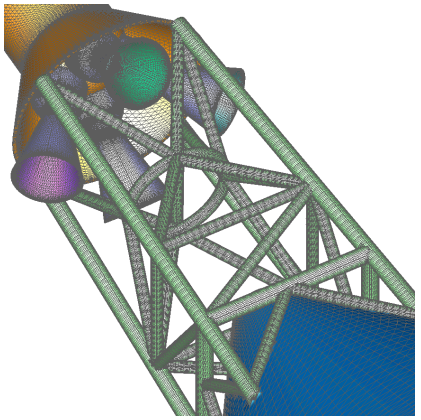


Cylinders: specify 1 diameter Frustums: specify 2 diameters

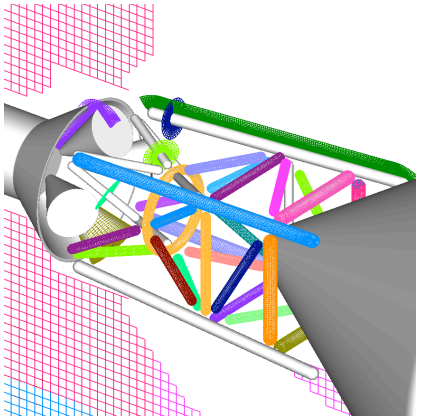
Same macro to create cylinder or frustum geometry and surface grids with various options for ends geometry

```
CreateCylGrids output_surface_grid_filename \
               iend (end geometry 1/2/3/4 1/2/3/4) \
               icap (cap grid off/on 0/1) \
               length \
               diameters [list diam1 diam2] \
               number_circumferential_points \
               stretching _ratio \
               grid_spacings [list ds_global ds_corner ds_axis] \
               centroid_coordinates [list xc yc zc] \
               axis_vector [list xa ya za]
```


RECENT APPLICATIONS THAT USED THE CYLINDER MACRO: Apollo Launch Escape System

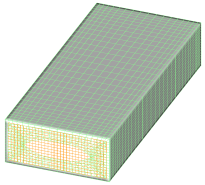


Surface triangulation for CART3D inviscid runs (intersecting cylinders)



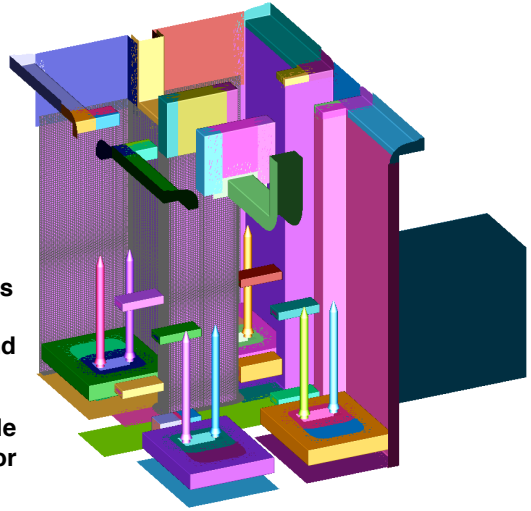
Surface and volume grids for OVERFLOW-2 viscous runs (non-intersecting cylinders)

CGT SCRIPT LIBRARY: RECTANGULAR_PRISM MACRO AND RECENT APPLICATION



Generate geometry and surface grids for rectangular prism given bounding box coordinates and grid spacings (O-grid main body with end caps)

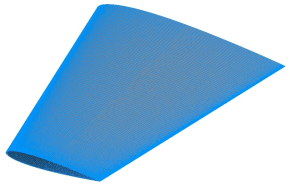
Recent application: Vehicle Assembly Building Interior



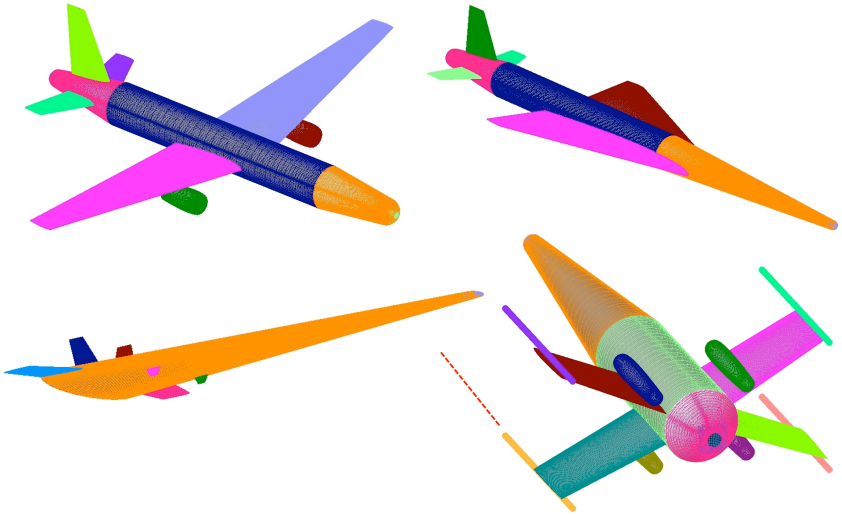
CGT SCRIPT LIBRARY: AIRFOIL_COMPONENT MACRO

- Generate geometry and surface grids for airfoil components with 2 or more sections
- Essential component for rapid conceptual analysis of aerodynamic vehicles (wings, tails, pylons, fins, sails, etc.)

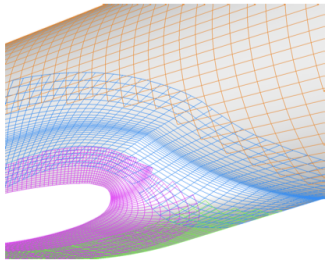
CreateAirfoilComponent number_of_sections \
name [list NACA NACA …] \
series [list 0012 0012 …] \
chord [list chord1 chord2 …] \
span [list span1 span2 …] \
le_sweep, dihedral, twist, … \
various grid parameters, … \
component_direction (x/y/z) \
output_grid_filename

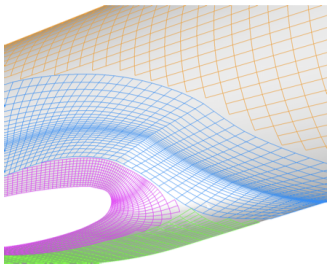


SAMPLE GEOMETRIES AND GRIDS GENERATED WITH THE CYLINDER AND AIRFOIL_COMPONENT MACROS



LOADS COMPUTATION USING HYBRID MESH



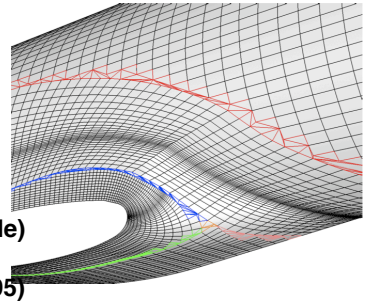


Motivation

Accurate accounting of overlapped zones needed for loads integration

Approach

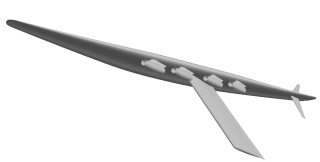
- Create hybrid mesh
- Integrate over non-overlapping quadrilaterals and triangles (mid-pt. rule)

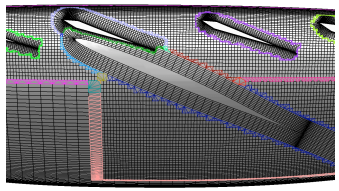


Software: MIXSUR 1.* (NASA Ames, 1995)

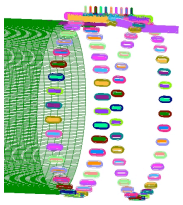
GRID SYSTEMS OF THE PAST AND PRESENT

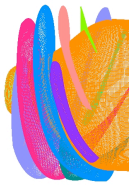
Mid 1990's: 5 - 10 grids, 5 - 10 million volume grid points

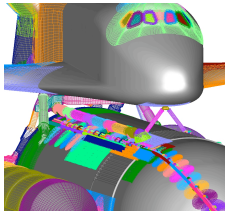


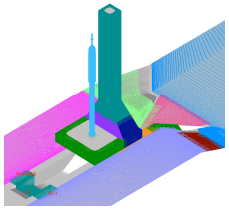


Today: 100+ grids, 100+ million volume grid points

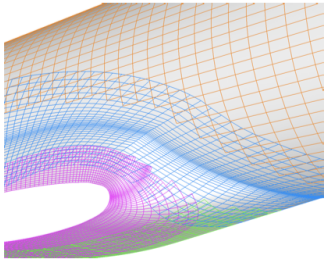


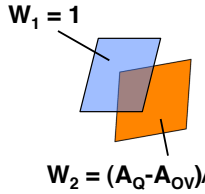






LOADS COMPUTATION USING WEIGHTED PANELS





A_Q = Area of quadrilateral
A_{OV} = Area of overlap

Approach
- Integrate over all quadrilaterals with weights $0 <= W_i <= 1$ for contribution from each quadrilateral (mid-pt. rule)
- Weights determined by polygon clipping and boolean subtraction

Software: POLYMIXSUR (Boeing, 2004), USURP (Penn State, 2005)

PROS AND CONS OF HYBRID MESH AND WEIGHTED PANELS APPROACHES

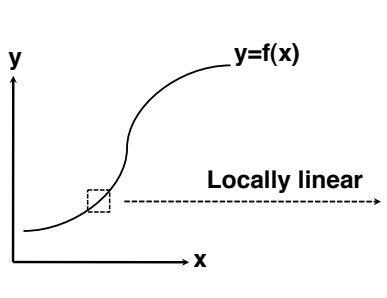
Hybrid Mesh Pros
- Exact integration of linear functions
- Visual check of errors (hybrid mesh)

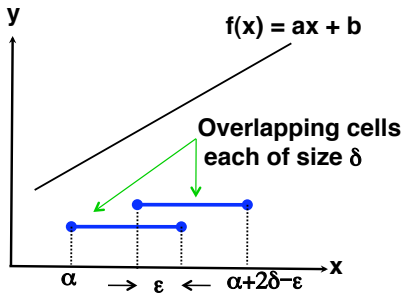
Hybrid Mesh Cons
- Not robust for large complex grid systems with wide mismatches in surface grid resolutions in overlapped zones
- May require input parameter iterations
- Execution speed (stencil search algorithm) scales with square of number of points on each surface subset

Weighted Panels Pros
- Always returns an answer
- No input parameter iterations required
- Fast relative to MIXSUR 1.*

Weighted Panels Cons
- Can only exactly integrate a constant function (accuracy is unclear beyond that)
- Difficult to assess error (absence of visual check like a hybrid mesh)

ONE-DIMENSIONAL LOCAL ERROR ESTIMATE ON OVERLAPPING CELLS





Functions to be integrated

$f(x) = -(p - p_{inf})$ pressure terms

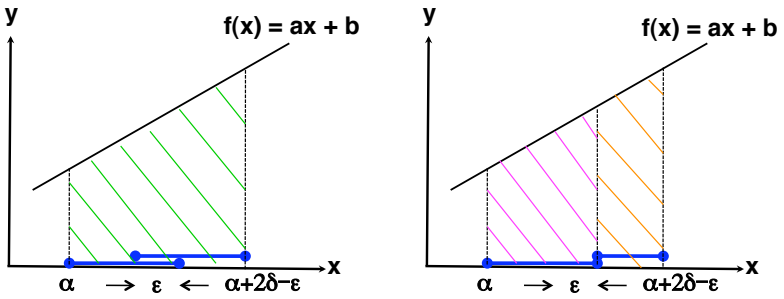
$f(x) = \tau_{ij} n_j$ viscous terms

ε = size of overlap with $0<\varepsilon<\delta$

Integrate from $x=\alpha$ to $x=\alpha+2\delta-\varepsilon$

Functional values known at node points

ONE-DIMENSIONAL LOCAL ERROR ESTIMATE FOR HYBRID APPROACH

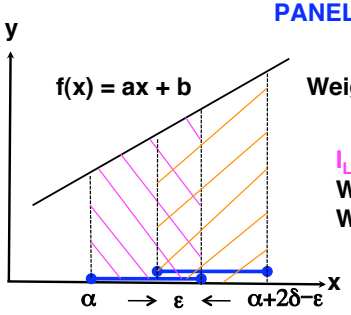


Exact integral = shaded area

Hybrid approach integral computed using mid-point rule on entire left cell and trimmed right cell
= exact integral for linear function

Local error for hybrid approach = 0 exactly

ONE-DIMENSIONAL LOCAL ERROR ESTIMATE FOR WEIGHTED PANELS APPROACH



Weighted panels integral $I_{WP} = I_L W_L + I_R W_R$

I_L, I_R = integrals on left and right cells
W_L = left cell weight = 1
W_R = right cell weight = $(\delta-\varepsilon)/\delta$

Let normalized overlap $\theta = \varepsilon/\delta$ = constant with $0<\theta<1$

Weighted panels integral error = I_{WP} – exact analytic integral
= $(-a/2)\,\theta(1-\theta)\delta^2$

- Magnitude of local error for weighted panels approach is
  - always > 0 for non-constant function
  - proportional to the local slope of the function
  - proportional to square of local cell size

MIXSUR VERSION 2.0

- Complete rewrite of MIXSUR using Fortran 90 with advanced data structures and dynamic memory allocation (version 2.0)

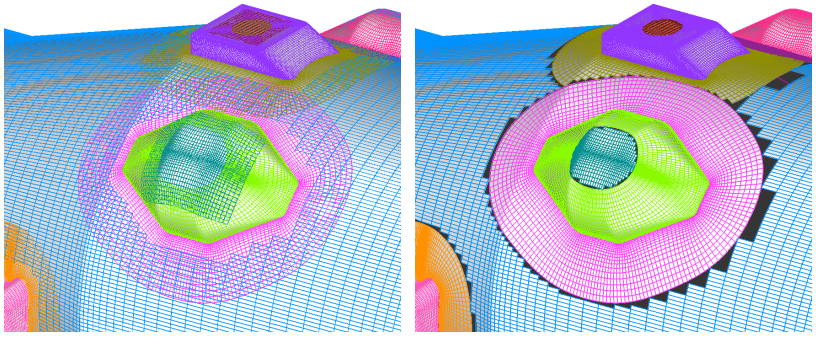
- New search algorithm to significantly reduce execution time

- New zipper grid (triangulation) algorithm to improve robustness

- Triangle vertex ordering (surface normal) determined by logical rather than floating-point test

- For each surface and component, report
  - A_q = surface area occupied by quadrilaterals
  - A_t = surface area occupied by triangles
  - a_q = fraction of total area occupied by quadrilaterals
  - a_t = fraction of total area occupied by triangles

HYBRID MESH CREATION PROCEDURE
1. Removal of quadrilateral cell overlap by blanking more vertices



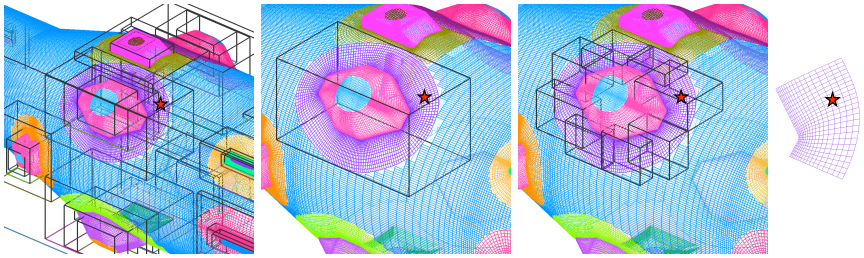
Vertex blanking from domain connectivity

Vertex blanking from overlap removal

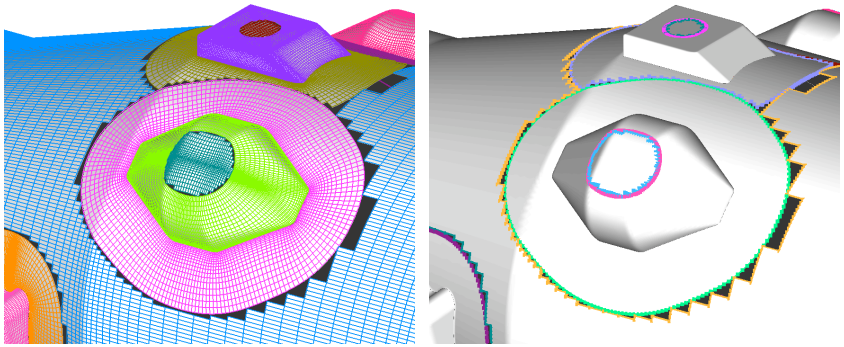
Tasks:
- For each vertex on each subset, search for donor stencil from neighboring subset, and mark vertices with donor stencils
- Between any pair of overlapping subsets, blank vertices from one subset or the other

NEW STENCIL SEARCH ALGORITHM

- Use surface subset bounding box as first filter
- Break each surface subset into smaller patches in index space
- Each patch has no more than N points in J and K directions
- Construct bounding box around each patch
- Given target point, identify patch bounding boxes that contain the point
- Do closest point test on small number of patches followed by stencil walk to get final stencil



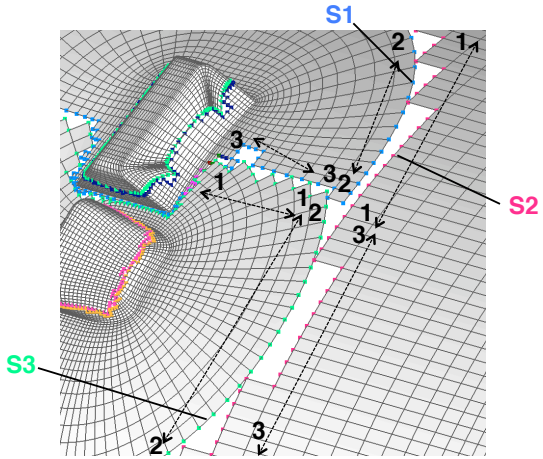
HYBRID MESH CREATION PROCEDURE
2. Creation of gap boundary strings



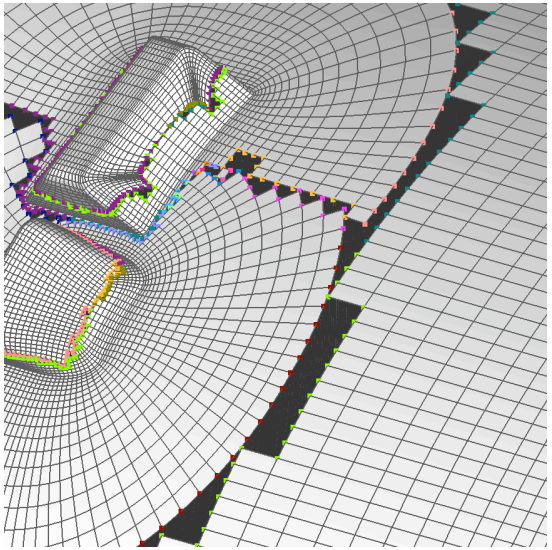
Tasks:
- Identify vertices and edges on gap boundaries
- Connect edges to form strings

STRING VERTEX MAP

For each vertex on each boundary string, determine and store
1. String ID of closest string
2. Vertex ID on closest string

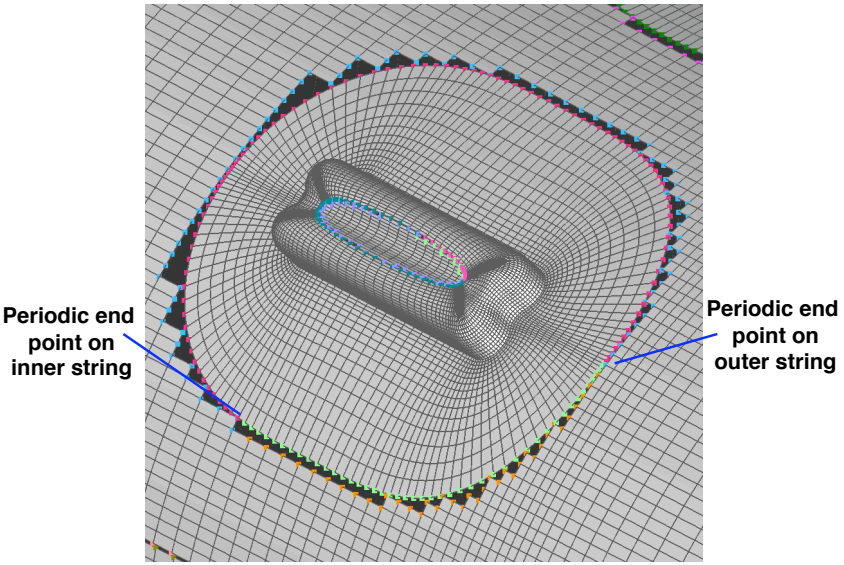


SUB-STRING SPLITTING
1. At vertex where neighboring string ID switches from one to another on vertex map



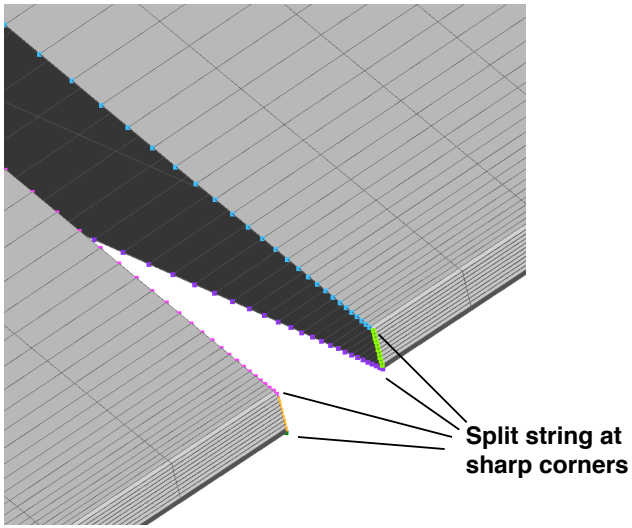
SUB-STRING SPLITTING

2. At vertex closest to periodic end point on neighboring string



SUB-STRING SPLITTING

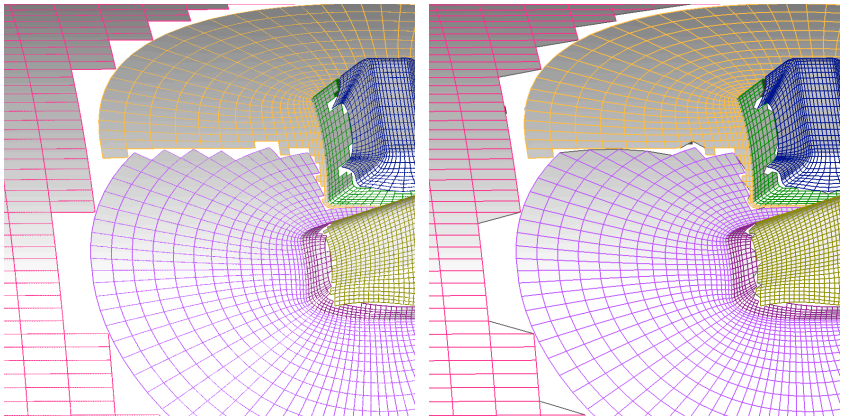
3. At vertex where angle deviation between local surface normals on each side exceeds specified threshold



HYBRID MESH CREATION PROCEDURE

3. Identify matching pairs of sub-strings and fill gap with triangles

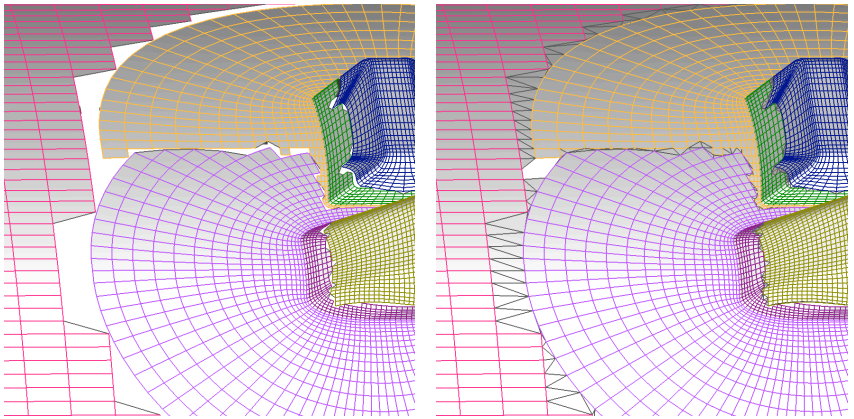
3.1 Self zipping on each sub-string



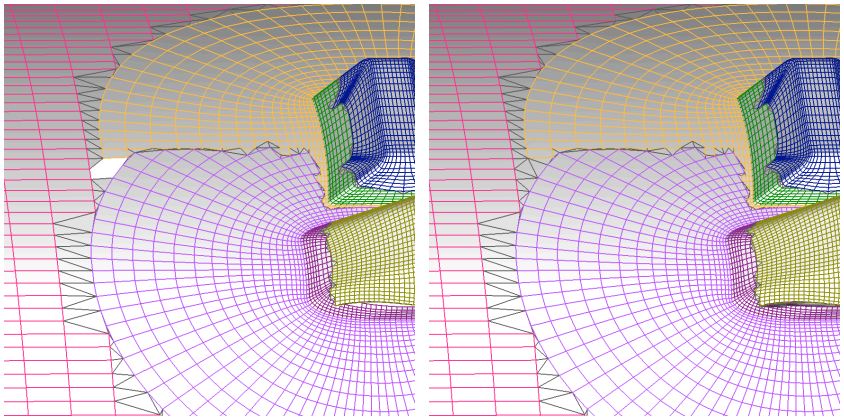
HYBRID MESH CREATION PROCEDURE

3. Identify matching pairs of sub-strings and fill gap with triangles

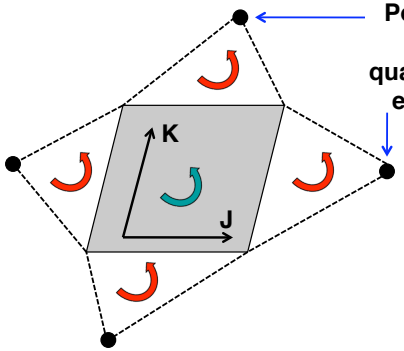
3.2 Cross zipping between matching pairs of sub-strings



HYBRID MESH CREATION PROCEDURE
3. Identify matching pairs of sub-strings and fill gap with triangles
3.3 Zipping of remaining polygonal gaps



ORDERING OF TRIANGLE VERTICES



- Each triangle has one edge that is shared with an existing quadrilateral of the original structured surface mesh
- Use directed edge of quadrilateral to determine ordering of vertices in triangle

SUMMARY

- Weighted panels method has a non-zero local error that scales with the local slope and the square of the grid spacing
- Use of index-space based bounding boxes for stencil searches leads to a ~40x speed up
- Robustness improved with sub-string splitting and triangle vertex ordering based on quadrilateral directed edge (~2x to ~10x speed up)
- Total speed up is about 80x to 400x

FURTHER ON-GOING WORK

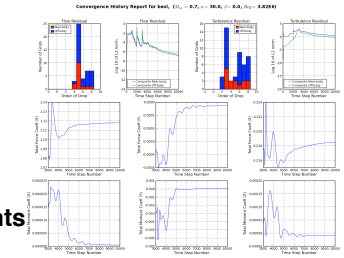
- More robust treatment of sub-string matching from two grids with unmatched geometric features
- Implementation of cross-over test between triangles and quadrilaterals
- More code testing on complex configurations: Space Shuttle
- Output of loads contribution from quadrilaterals and triangles separately

SOLUTION MONITORING AND REPORTING TOOL (OVERSMART)

- Executive summary view of histories of

  - flow variables residuals
  - turbulence model variables residuals
  - component forces and moments

  for large number of grids and components



- Needed for fast and automatic assessment of solution convergence for steady state cases and unsteady behavior for time accurate cases

- Current status: command line, post-processing mode
  (see David Kao's talk for details)

- Work in progress: GUI and dynamic monitoring modes